# DevOps Runbook

|  |  |
| --- | --- |
| **Runbook name** | **CommScope DevOps Runbook** |
| **Runbook description** | Detailed description of the aws services, github, Jenkins pipeline and  terraform |
| **Contents** | 1. Git Repositries 2. Environments    1. Dev and Pre-Prod account services    2. Production account services 3. Jenkins pipeline    1. Jenkins credentials 4. Terraform |

## 1. ****Git Repositories****

GitHub repositories are utilized to store the code. Here is a list of dedicated repositories for each team,

* mi-inbound-db: Used by DB team
* mi-inbound-platform: Used by platform team
* mi-inbound-web: Used by UI team
* mi-inbound-foundation: Used for the devops related activities.

## 2. Environments

Development and Pre-Prod environments are in AWS account **692425240801**. And Production environment is in **867867082494**.

**2.1 Dev and Pre-Prod account services (692425240801)**

The services used in the non-prod account is listed as below,

**• S3**

1. mi-inbound-reports-data-dev: Is used by platform team in the development environment for report sorting purpose.
2. mi-inbound-reports-data-preprod: Is used by platform team in the pre-prod environment for report sorting purpose.
3. mi-inbound-dashboard-serverless: Is used by platform team in the development environment for serverless deployment.
4. inbound-crm-reporting-website-dev: Is used by UI team in the development environment for web deployment.
5. inbound-crm-reporting-website-preprod: Is used by UI team in the pre-prod environment for web deployment.

**• Cloudfront**

1. E1ALZ0BIQZCVOJ: It is created in the dev environment for accessing the inbound-crm-reporting-website-dev bucket.
2. E1YM70XLN7OHSM: It is created in the dev environment for accessing the inbound-crm-reporting-website-preprod bucket.

**• Amazon Redshift**

1. crm-inbound-redshift-clustrer-dev:

* It is created in the pre-prod environment. It consists of 2 number of nodes and the node type is dc2.large with 320GB storage.
* VPC associated is vpc-0f90b88c73d0d854c. Subnet is crm-inbound-dev-cluster-subnet-group.
* Associated parameter group is wlm-inbound-rpt-dev.
* Associated IAM roles are, crm-inbound-S3\_redshift-clustrer-dev\_role, dms-access-for-endpoint, MSILInboundRedshiftKinesisRole and Redshift\_Kinesis\_role.
* Automated snapshot retention period is 1 day, and manual snapshot retention period is indefinite.
* awsredshift-crm-inbound-redshift-cluster-preprod-CPUUtilization-ALL alarm is associated with the cluster to get the notification if the CPU utilization of the cluster if exceeds 80%.

1. crm-inbound-redshift-cluster-dev-new:

* It is created in the dev environment. It consists of 2 number of nodes and the node type is dc2.large with 320GB storage.
* VPC associated is vpc-0f90b88c73d0d854c. Subnet is crm-inbound-dev-cluster-subnet-group.
* Associated parameter group is wlm-inbound-rpt-dev-new.
* Associated IAM roles are, dms-access-for-endpoint, MSILInboundRedshiftKinesisRole and Redshift\_Kinesis\_role.
* Automated snapshot retention period is 1 day, and manual snapshot retention period is indefinite.
* awsredshift-crm-inbound-redshift-cluster-dev-new-CPUUtilization-ALL alarm is associated with the cluster to get the notification if the CPU utilization of the cluster if exceeds 80%.

**• Kinesis**

1. inbound\_email\_pre\_prod\_stream
2. email-reporting-kinesis-sync-stream-dev
3. wastecall-reporting-kinesis-sync-stream-preprod
4. prospect\_reporting\_pre\_prod\_stream
5. wastecall-reporting-kinesis-sync-stream-dev
6. prospect\_reporting\_stream

**• EC2 instance**

1. inbound\_crm\_reporting\_node: It is a t2.medium instance used by the platform team for the deployment purpose.
2. crm-inbound-dev-bastion: It is a t3.large instance used by the DB team to connect to the redshift cluster.

**• SNS**

1. crm-inbound-redshift-sns-dev: It is used to get alert notifications from redshift and DMS for the cpu utilization and DMS state changes.

**• Lambda**

1. mi-inbound-dashboard-service-dev-dumpReport – Triggered by api gateway for dev environment.
2. mi-inbound-dashboard-service-dev-reportFilter – To get filters for the reports. To be inflated the UI for dev environment
3. mi-inbound-dashboard-service-dev-dumpReportFilter – Filter based on the case type filter for dev environment
4. mi-inbound-dashboard-service-dev-reportRequest – To requesting report for dev environment
5. mi-inbound-dashboard-service-dev-bookmarkConfig– To manage bookmarks for dev environment.
6. mi-inbound-dashboard-service-dev-lastUpdatedData – To get the last update details for dev environment.
7. mi-inbound-dashboard-service-dev-emailReportRequest– Request for email report for dev environment.
8. mi-inbound-dashboard-service-preprod-reportFilter– To get filters for the reports. To be inflated the UI for pre-prod environment.
9. mi-inbound-dashboard-service-preprod-dumpReportFilter – Filter based on the case type filter for pre-prod environment.
10. mi-inbound-dashboard-service-preprod-emailReportRequest – Request for email report for pre-prod environment.
11. mi-inbound-dashboard-service-preprod-bookmarkConfig – To manage bookmarks for pre-prod environment.
12. mi-inbound-dashboard-service-preprod-dumpReport – Report related information for pre-prod environment.
13. mi-inbound-dashboard-service-preprod-lastUpdatedData – To get the last update details for pre-prod environment.
14. mi-inbound-dashboard-service-preprod-reportRequest – To requesting report for pre-prod environment.
15. mi-inbound-socket-service-dev-broadcast – To generate report and broadcast for dev environment.
16. mi-inbound-socket-service-dev-downloadConnect – To connect the socket services for dev environment.
17. mi-inbound-socket-service-dev-emailReport – Generate email reports for dev environment.
18. mi-inbound-socket-service-preprod-broadcast – To generate report and broadcast for pre-prod environment.
19. mi-inbound-socket-service-preprod-downloadConnect – To connect the socket services for pre-prod environment.
20. mi-inbound-socket-service-preprod-emailReport – Generate email reports for dev environment.
21. mi-inbound-socket-service-dev-mosReport – Mos Report generation for dev environment.
22. mi-inbound-dashboard-service-dev-mosReportRequest - Mos Report generation for dev environment.

**• API gateway**

1. dev-mi-inbound-dashboard-service
2. preprod-mi-inbound-dashboard-service
3. mi-inbound-report-socket-service-dev
4. mi-inbound-report-socket-service-preprod

**2.2 Production account services (867867082494)**

**• S3**

1. inbound-reports-data: Is used by platform team in the production environment for report sorting purpose.
2. inbound-reporting-website: Is used by UI team in the production environment for web deployment.
3. crm-reporting-serverless-production: It is used by platform team for the serverless deployment.
4. inbound-crm-rpt-DB-data-archive: It is used by the Db team to Archive raw data from Redshift to S3 bucket.

**• Cloudfront**

1. E1ARY9MP5HB1GP: It is used as the distribution for inbound-reporting-website bucket.

**• Amazon Redshift**

1. crm-inbound-redshift-cluster: It consists of 4 number of nodes and the node type is dc2.large with 640GB storage. VPC associated is [vpc-04c12635813e72926](https://console.aws.amazon.com/vpc/home?region=ap-south-1#VpcDetails:VpcId=vpc-04c12635813e72926). Subnet is crm-inbound-dev-cluster-subnet-group. Automated snapshot retention period is 1 day, and manual snapshot retention period is indefinite.

**• Kinesis**

1. inbound-crm-wastecall-reporting-kinesis-sync-stream
2. inbound-crm-email-docdb-sync-stream
3. prospect\_reporting\_stream

**• Lambda**

1. mi-inbound-dashboard-service-prod-reportFilter - To get filters for the reports. To be inflated the UI.
2. mi-inbound-dashboard-service-prod-dumpReportFilter – Filter based on the case type filter
3. mi-inbound-dashboard-service-prod-emailReportRequest – Request for email report.
4. mi-inbound-dashboard-service-prod-bookmarkConfig – To manage bookmarks.
5. mi-inbound-dashboard-service-prod-dumpReport – Report related information
6. mi-inbound-dashboard-service-prod-lastUpdatedData -To get the last update details.
7. mi-inbound-dashboard-service-prod-reportRequest - To requesting report.
8. mi-inbound-socket-service-prod-broadcast - To generate report and broadcast.
9. mi-inbound-socket-service-prod-downloadConnect - To connect the socket services.
10. mi-inbound-socket-service-prod-emailReport - Generate email reports.

**• API gateway**

1. dev-mi-inbound-dashboard-service
2. preprod-mi-inbound-dashboard-service
3. mi-inbound-report-socket-service-dev
4. mi-inbound-report-socket-service-preprod

**2.3 Non-Production AWS account user groups, roles and policies**

### ****a. User groups****

### 1. MSIL\_HM\_Inbound\_DB and MSIL\_HM\_Inbound\_DB\_2: The group includes the DB team. The permissions included are as listed below,

### • AmazonEC2ReadOnlyAccess

### • AmazonEventBridgeFullAccess

### • AmazonKinesisReadOnlyAccess

### • AmazonRedshiftAllCommandsFullAccess

### • AmazonRedshiftFullAccess

### • AmazonRedshiftQueryEditor

### • AmazonRedshiftQueryEditorV2FullAccess

### • CloudWatchReadOnlyAccess

### • IAMUserChangePassword

### • AWSGlueConsoleFullAccess

### • AWSLambda\_FullAccess

### • MSILInboundDBPolicy – A custom managed policy

### 2. MSIL\_HM\_Inbound\_Developer: The group includes the developer permissions. The permissions are as listed below,

### • AmazonEC2ReadOnlyAccess

### • AmazonS3FullAccess

### • AmazonRedshiftFullAccess

### • CloudWatchFullAccess

### • IAMUserChangePassword

### • CloudFrontFullAccess

### • AWSLambda\_FullAccess

### • AmazonSNSFullAccess

### • AmazonSQSFullAccess

### • MSILInboundDeveloperPolicy – A custom managed policy

### ****3. Jenkins pipeline****

### Pipeline is created in Jenkins for the automation purpose.

### mi-inbound-platform-dev-job: The job is created for the automation of the platform deployments. It consists of five stage such as,

### - Checkout: The code is checked out from the github repository.

### - Build: Code building takes place at this stage

### - Sonarqube analysis and quality gate: In this stage, the code quality is checked, and the sonar report will be generated

### - Deploy: Deployment action takes place at this stage.

### - Post build action: This stage includes the notification part in which mail is sent to the developers with the log file attached.

### mi-inbound-web-dev-job: The job is created for the automation of the web deployments. It consists of five stage such as,

### - Checkout: The code is checked out from the github repository.

### - Build: Code building takes place at this stage

### - Sonarqube analysis and quality gate: In this stage, the code quality is checked, and the sonar report will be generated

### - Deploy: Deployment action takes place at this stage.

### - Post build action: This stage includes the notification part in which mail is sent to the developers with the log file attached.

### terraform: The job is to aws services using terraform.

### 3.1 Jenkins credentials:

### Required credentials are stored in Jenkins that are utilized during the build.

### Reporting-Jenkins: This credential ID contains the access key and secret key of the non-prod aws account.

### git\_cred: This credential ID contains the access details of github.

### ****4. Terraform****

### Terraform is used to create the AWS infrastructure as a code. Services such as redshift, kinesis, sns, s3 and cloudfront are created. The code is stored in the mi-inbound-foundation repository with branched named after the services.